

## Recipe App version 1.0

**Application-** Spring Boot

**IDE-** Eclipse

**Front End-** Bootstrap

**Data-** Raw data taken from internet












### Use Case 1

Input :Read recipe.json from resources

Output: Sorted list of all recipes. Lambda expression used for sorting based on recipe title.

### Result

1. Refer - recipeList() function in Homepage.java controller.  
Scriptlets have been used to refer Utility.java class under util package to read and display sorted list of recipes in home.jsp.
2. Enter <http://localhost:8080/> in the browser to view the result.

Recipe List			
Recipes	Link	Ingredients	Images
Barbecued Grilled Portobello Sandwich	<a href="#">Click Here</a>	[mushrooms, barbecue sauce, sandwich rolls, lettuce, tomato]	
Barbecued Grilled Portobello Sandwich	<a href="#">Click Here</a>	[mushrooms, barbecue sauce, sandwich rolls, lettuce, tomato]	
Beef & Portobello Mushroom Stroganoff	<a href="#">Click Here</a>	[mushrooms]	
Blue Ribbon Meatloaf	<a href="#">Click Here</a>	[onions]	
Boiled Ham	<a href="#">Click Here</a>	[ham, onions]	
Broccoli Casserole Recipe	<a href="#">Click Here</a>	[onions]	
Bruschetta With Roasted Garlic and Cherry Tomatoes	<a href="#">Click Here</a>	[garlic, italian bread]	
Bryannas Vegan Dulce De Leche Recipe	<a href="#">Click Here</a>	[rice]	
Cheesy Hot Dog Rice	<a href="#">Click Here</a>	[rice]	
Chicken Salad With Rice Noodles	<a href="#">Click Here</a>	[rice]	
Chicken With Orange Sauce Recipe	<a href="#">Click Here</a>	[rice]	

3. Junit Test Case

Junit test case implemented to test that recipe.json should not be empty.

Refer testReadRecipeJson() in UtilityTest.java.

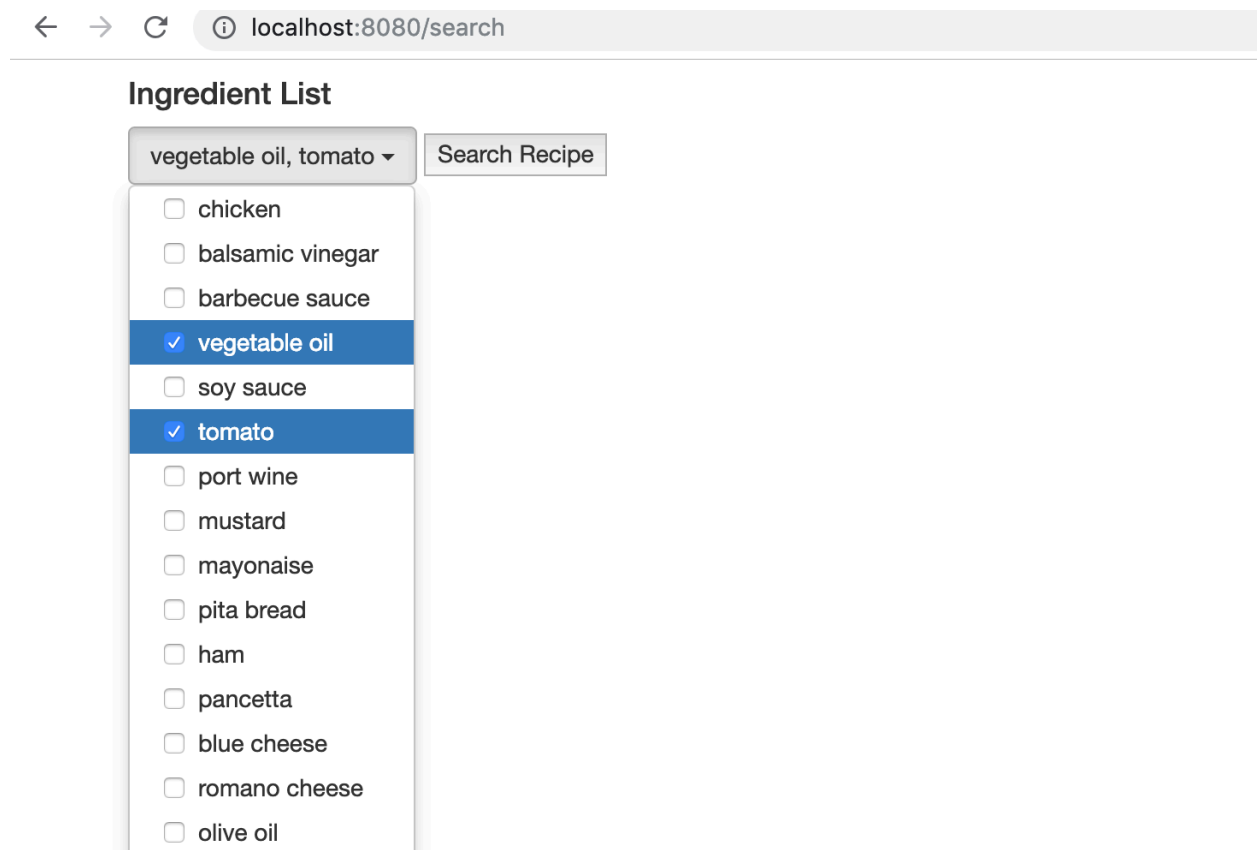
## Use Case 2

Input – Read ingredients from recipe.json

Output- View having list of all ingredients from recipe.json with multi-select functionality and search button.

### Result

1. Refer listIngredients() function in Homepage.java controller.  
JSTL and EL have been used to understand it's advantage over scriptlets in *ingredients.jsp*
2. Enter <http://localhost:8080/search> in the browser to view the result.



← → ↻ ⓘ localhost:8080/search

### Ingredient List

vegetable oil, tomato ▾ Search Recipe

- ☐ chicken
- ☐ balsamic vinegar
- ☐ barbecue sauce
- ☒ vegetable oil
- ☐ soy sauce
- ☒ tomato
- ☐ port wine
- ☐ mustard
- ☐ mayonaise
- ☐ pita bread
- ☐ ham
- ☐ pancetta
- ☐ blue cheese
- ☐ romano cheese
- ☐ olive oil

3. Junit Test Case  
Junit test case implemented to test that ingredients list should not be empty.  
Refer testReadIngredients() in UtilityTest.java.

### Use Case 3

Input- Selected Ingredients from use case2.

Output- List of recipes which consists of all the given input ingredients.

For eg.,

1. If input is ["onions"], then it should list down all the recipe's which has onions in the list of ingredients
2. If input is ["onions", "mushrooms"] then it should list down all the recipe's which contains onions and mushrooms in the list of ingredients

### Result

1. Refer displayRecipe() function in Homepage.java controller. searchRecipe() function of Utility.java has been used to search recipe based on the selected ingredients. Again JSTL and EL have been used to display recipe. There will be blank screen in case of no recipe of selected ingredients.
2. Enter <http://localhost:8080/list> in the browser to view the result.

## Recipe List

Green Bean Casserole  
Boiled Ham  
Pulled Chicken Sandwiches (Crock Pot)  
Blue Ribbon Meatloaf  
Grilled Portobello Mushrooms  
Spaghetti with Clams & Corn  
Creamy Scrambled Eggs Recipe Recipe  
Broccoli Casserole Recipe  
Crock Pot Caramelized Onions  
Grilled Chipotle Salmon With Pineapple Cilantro Rice  
Roast Chicken with Rosemary

3. Junit Test case  
Junit test case implemented to check the list of recipes of selected ingredients  
e.g. {"onions", "butter"}. Refer testSearchRecipe() in UtilityTest.java.

## Recipe App Version 1.1

In next version data from recipe.json will be stored in the database and rest of the manipulation will be done using database.

Spring with Hibernate will be used.

Database PostgreSQL

Functionality will be included in Dao, Service packages